CERT-EU Ticket Queue Classification

Machine Learning Solution Report

Marko Jovanovic

August 6, 2025

1. Problem Approach

The goal of the task is to automatically classify security tickets from CERT-EU into 7 different queues. The aim is to minimize the human workload and maintain the high reliability of the classification.

Dataset

To approach the problem concisely, an analysis of the dataset had first to be made. In particular, what are the specificities of the different classes or is there any ambiguity between two different operational queues (e.g, DFIR::phishing vs DFIR::incidents). Finally, the distribution of each queue had to be assessed (see Table 1). What has been noticed is that the dataset was imbalanced but not hardly imbalanced; therefore, a class weighting would be applied to make sure that all the classes are considered equally. In the next step, because this is a free-text classification problem, additional features were extracted to help the model distinguish between ambiguous categories. These features provide extra context, making it easier for the model to correctly classify the different types of tickets. The feature set includes temporal, security-related, and more general indicators, as well as a clean version of the text (see Section 2). In terms of preprocessing, the text is first lowercased, HTML tags, URLs, and email addresses are removed, and only alphabetic characters are kept. Stopwords are filtered out and lemmatization is applied, so that different forms of a word are treated as equivalent (for example, "attacks" and "attack" are mapped to the same token). The list of security-related keywords was generated with the help of ChatGPT, simply to provide a broad set of commonly used cybersecurity terms. Based on all these assumptions, the extracted features leverage both the unstructured text (title and content) and structured metadata (timestamp, attachment indicators, security keywords, etc.).

Assumptions Summary

- Hybrid data: Each ticket contains both free-text (title, content) and structured metadata (sender, timestamp, attachments, etc.).
- Class imbalance: Some queues are more frequent (see Appendix for distribution).
- Security-specific language: CVE IDs, IOCs, and phishing lingo require domain-aware preprocessing.
- Operational reliability: Reliable auto-routing is important; truly ambiguous cases should be reviewed by a human.

2. Model Architecture

Model selection was guided by several practical and technical considerations. Given the need to process both unstructured text and structured metadata, the architecture is based on a hybrid design. The text branch uses a pre-trained RoBERTa transformer [3], fine-tuned on the ticket titles and contents. This provides a contextual embedding for each ticket by extracting the [CLS] token from the last hidden state [1].

Text and Numerical Feature Networks

Textual data is processed using the RoBERTa encoder, while structured features—such as sender domain, temporal attributes, URL statistics, and security flags—are scaled and passed through a dedicated two-layer feedforward neural network. Batch normalization, ReLU activations, and dropout are used to ensure stable learning and avoid overfitting.

Attention-Based Fusion

To effectively combine information from both modalities, an attention-based fusion layer is applied. Both text and numerical embeddings are projected into the same hidden dimension, and multi-head attention [4] is used so that the textual embedding can attend to the numerical one. This enables the model to learn more complex interactions between text and structured data, beyond what simple concatenation would offer. A residual connection and layer normalization are applied, followed by a final projection to prepare the fused representation for classification. The result is then passed to a standard classification head to obtain the logits for each queue.

Imbalance Handling and Confidence-Based Routing

Given the class imbalance in the data, a class-weighted cross-entropy loss is used to encourage the model to pay attention to underrepresented queues. During inference, the softmax probabilities output by the model are used to implement confidence-based routing: tickets with high-confidence predictions are auto-routed, borderline cases are flagged for human review, and ambiguous tickets are triaged manually. This approach maintains automation while reducing risk in uncertain cases.

Training Strategy

The model is trained using AdamW with a batch size of 16 and a learning rate of 2×10^{-5}. An automatic learning rate finder was first tested, but it did not produce conclusive results; for this reason, the final learning rate was selected based on values recommended in the literature [2]. Early stopping is applied based on validation performance, the convergence being achieved at the fifth epochs on this dataset. To ensure robust generalization and to assess model stability, five-fold stratified cross-validation is used throughout. The architecture is implemented using the Huggingface Transformers library [5] and is shown in Appendix 1.

Architecture Summary

- Uses a pre-trained RoBERTa transformer for text encoding, fine-tuned on ticket data.
- Processes structured metadata through a dedicated two-layer feedforward network.
- Combines text and structured features using an attention-based fusion layer.
- Applies class-weighted cross-entropy loss to address class imbalance.
- Implements confidence-based routing to ensure reliability by directing high-confidence tickets to auto-routing and sending low-confidence tickets for human review.
- Trained using established transformer fine-tuning practices and the Huggingface Transformers library.

3. Performance Evaluation

Validation Strategy

5-fold stratified cross-validation has been used to evaluate model performance. The goal is to ensures each fold preserves the class distribution of the dataset. This helps to see how well the model generalizes and how stable it is across different splits. Moreover, a qualitative approach has been done where GPT-4o was used to generate queue predictions for the test dataset, and these predictions were taken as a sort of reference point. The mismatches between the model's predictions and those of GPT-4o were then manually reviewed. The intuition is that if the model consistently outperforms or matches GPT-4o knowing how strong transformer models already are for classification it's a good sign that the solution is robust.

Metrics

The metrics being reported are: accuracy, macro-averaged F1, weighted F1 score. Also the following metrics were added: Per-class precision, recall and F1 and confidence-based routing states.

Results

Based on the results, the model performs really well on the dataset. However, we need to keep in mind that the dataset is small and that we need to have a bigger one to really be able to conclude how well does it perform in a real-world scenario. Nonetheless, by doing a qualitative and quantitative analysis, we can assume that the model seems to be able to automate a lot of the queues with a confidence of 90% which already shrinks a considerable amount of tickets to process for EU employees. (See Appendix for detailed metrics and queue distribution.)

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186, 2019.

[2] Xueqing Liu and Chi Wang. An empirical study on hyperparameter optimization for fine-tuning pre-trained language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2286–2300, Online, 2021. Association for Computational Linguistics.

[3] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. In *arXiv preprint arXiv:1907.11692*, 2019.

[4] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

[5] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing, 2019.

A.1 Queue-distribution Training set

Queue	Count	Percentage
CTI	228	24.4%
DFIR::incidents	171	18.3%
DFIR::phishing	164	17.6%
SMS	124	13.3%
OFFSEC::CVD	108	11.6%
Trash	71	7.6%
OFFSEC::Pentesting	68	7.3%

Table 1: Queue distribution in the CERT-EU ticket dataset.

A.2 Feature List

Feature	Description
Cleaned text	Title, content, and combined string after lowercasing, stopword removal, etc.
Title/content length	Number of characters in title/content
Title/content word count	Number of words in title/content
URL statistics	Number of URLs, disguised URLs (hxxp), unique domains, suspicious TLDs
Security flags	Presence of CVE IDs, version numbers, number of attachments
Email/domain metadata	Sender's domain, internal (EU) flag, domain TLD
Temporal features	Hour of day, day of week, weekend, business hour flag
Security keyword count	Count of security-specific keywords in text

Table 2: Features extracted for model input.

B.1 Model Architecture Pipeline

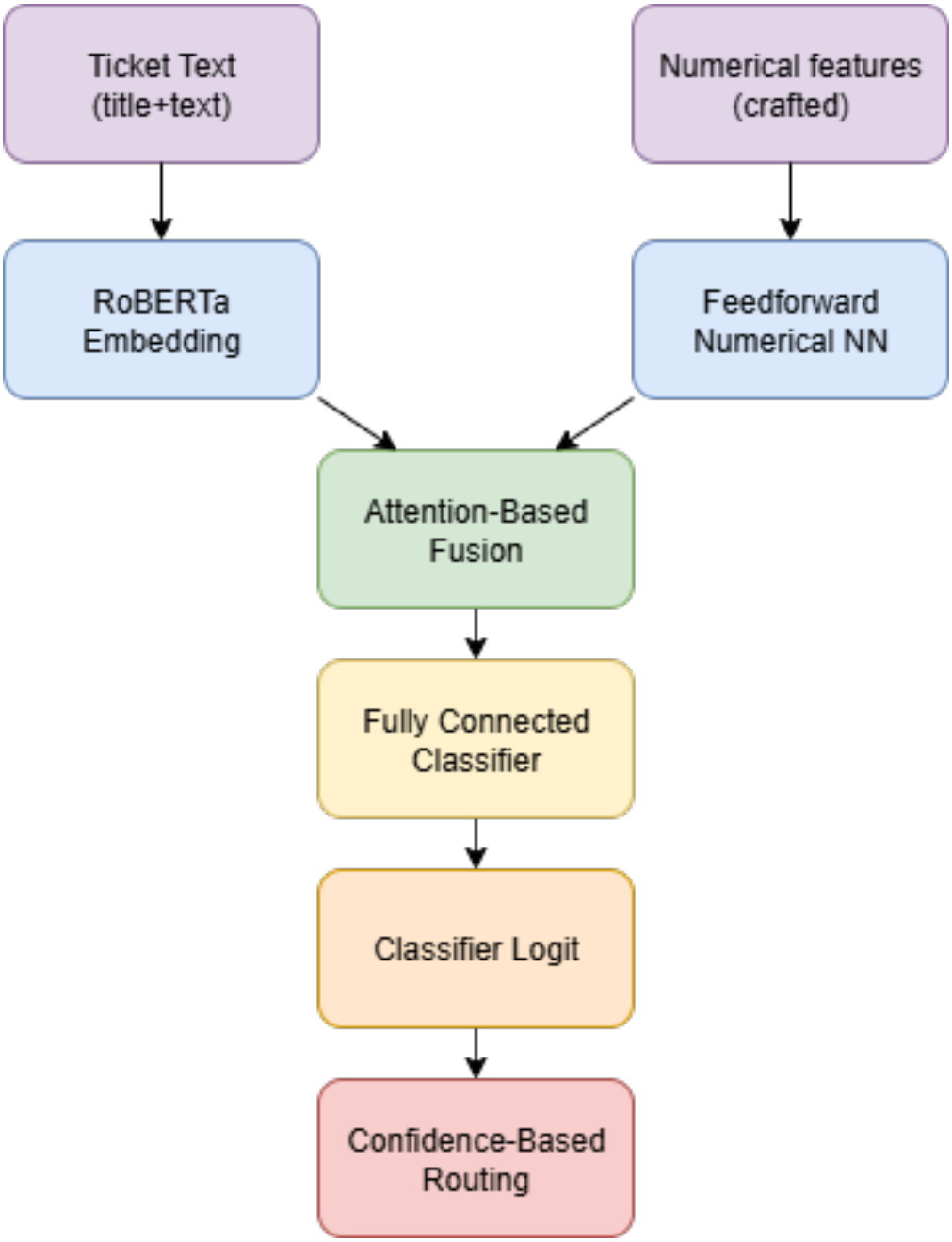

Figure 1: Hybrid RoBERTa model with attention-based fusion, class-weighted loss, and confidence-based routing for CERT-EU ticket classification.

C.1 Cross-Validation Performance

Metric	Mean	Std. Dev.
Accuracy	0.9861	0.0064
Macro F1	0.9880	0.0038
Weighted F1	0.9860	0.0065

Table 3: Cross-validation results (5 folds) on the training set.

C.2 Confidence-Based Routing

Routing Type	Percentage (%)
Auto-route ($p \geq 0.9$)	94.8
Human verify ($0.6 \leq p < 0.9$)	4.7
Manual triage ($p < 0.6$)	0.5

Table 4: Distribution of tickets by confidence threshold for automated routing.

C.3 Confusion Matrix of Cross-Validation

	CTI	DFIR::inc.	DFIR::phish	CVD	Pentest	SMS	Trash
CTI	45.0	0.4	0.0	0.2	0.0	0.0	0.0
DFIR::inc.	0.8	32.8	0.6	0.0	0.0	0.0	0.0
DFIR::phish	0.0	0.2	32.6	0.0	0.0	0.0	0.0
CVD	0.0	0.0	0.0	21.6	0.0	0.0	0.0
Pentest	0.0	0.0	0.2	0.0	13.2	0.2	0.0
SMS	0.0	0.0	0.0	0.0	0.0	24.8	0.0
Trash	0.0	0.0	0.0	0.0	0.0	0.0	14.2

Table 5: Mean confusion matrix over cross-validation folds. Rows = true labels, columns = predicted labels.

C.4 Routing Efficiency by Queue

Queue	Auto	Verify	Manual	Tot	Auto(%)	Acc.
CTI	220	8	0	228	96.5	0.99
DFIR::inc.	168	2	1	171	98.2	0.99
DFIR::phish	164	0	0	164	100.0	0.99
CVD	99	8	1	108	91.7	1.00
Pentest	63	5	0	68	92.6	1.00
SMS	123	1	0	124	99.2	1.00
Trash	67	2	2	71	94.4	1.00

Table 6: Routing efficiency and auto-route accuracy by queue type.